

Klasser

I min uppgift kommer jag jobba med en mängd klasser. Jag tänker beskriva de tre jag anser som viktigast i min uppgift.

class Player:

Player klassen representerar spelaren och kommer hantera allt som spelaren har, exempelvis items, antal pokemons på sitt lag osv.

Attributes:

- str - name
- list - team # lista med pokemon objekt som är i spelarens lag
- int - potion # antal potions man har
- int - pokeball # antal pokebollar man har

Methods:

```
def __init__(self):  
    skapar en spelare  
  
def __str__():  
    skriva ut spelaren på en mer läsbar format  
  
def actions(self):  
    ska innehålla vad en spelare kan göra under en encounter, kommer  
    returnerar en av attack, flee, items eller swap  
  
def movement(self):  
    ta inputs så spelaren rör på sig  
  
def attack(self):  
    anropas för att ens pokemon ska attackera  
  
def items(self):  
    för att spelaren ska välja/använda items  
  
def swap(self):  
    byta ut pokemon
```

class Pokemon

Pokemon klassen representerar individuella pokemon.

Attributes:

```
class Stats - stats # hanterar alla stats
class leveling - leveling # hanterar alla leveling metoder
class moves(list) - moves # lista av alla attacker en pokemon har
bool - fainted
int - baseAtk
int - baseHp
int - baseDef # Dessa tre används för kalkulationer vid level up osv
```

Methods:

```
def __init__(self):
    skapar en pokemon

def __str__():
    skriva ut pokemonen på en mer läsbar format

def gainExp(self):
    ökar pokemonens exp

def attack(self, other):
    tar in en annan pokemon objekt och gör skada på den

def damaged(self):
    minskar hp
```

class Encounter

Jag kommer att strukturera encounters som klasser istället för funktioner. Jag tycker att det är lättare då att hålla koll på olika tillstånd, hantera en turn-based system och skapa instanser av encounters.

Attributes:

```
class Player - player
class Pokemon - enemy
int - turn
```

Methods:

```
def __init__(self):
    skapar en encounter

def start(self):
    startar en intern loop i encounter

def end(self):
    avslutar encounter
```

```
def checkEndCondition(self):  
    kollar olika conditions för att anropa end när det behövs  
  
def nextTurn(self):  
    nästa tur
```

Funktioner

def startMenu()

skriver ut en meny för att användaren ska kunna starta eller exempelvis importera ett lag osv

def exportTeam()

spara sitt nuvarande lag i en ny textfil för framtida användning

def fileImport(file)

läser en fil och skapar pokemon objekt från den datan

def inGameMenu()

meny som spelaren använder när de börjar spelet

def encounter(player, pokemon)

skapar en encounter instans

def generatePokemon()

skapar en pokemon instans med random stats

def main()

cool beans 😎

Datastruktur



Varje pokemon kommer ha attributer som är andra klasser såsom stats, leveling och moves. Varje player instans kommer ha flera pokemons som lagras i en lista och är spelarens lag. En encounter tar en player instans och en pokemon instans som motståndare.

Algoritm

- Programmet läser in en fil på alla pokemons
- Kör startmeny i loop
- Om spelaren väljer att spela så körs inGameMeny istället och startar en ny loop
- Spelaren får gå runt, slåss i encounters, och spela spelet
- Spelet körs tills användaren ger någon input för att spara/avsluta

Tidsplan

Arbetet kommer att jobbas med kontinuerligt under flera veckor. Planen är att åtminstone jobba minstän få in 10 timmars arbete per vecka för att ligga inom 80 timmars. Lättast för mig att kunna uppfylla det målet är att köra 1 timme varje dag på vardag och 2-3 timmar på lördag och söndag. Men jag kommer även jobba mer vid vissa arbetstillfällen och mindre vid

andra beroende på andra uppgifter jag har då. Såklart kommer jag även kunna lägga ner mer tid per dag under jullovet när jag känner för det. Just nu är det vecka 49 och arbetet ska vara inlämnad vecka 5. Det är svårt att säga exakt vad jag kommer göra varje vecka men jag kommer sätta några lätta deadlines. Mitt arbete kan delas in i 4 delar som är ungefär lika mycket arbete, player och pokemon klasser, encounters, meny och gui. Jag ska försöka lägga ungefär lika mycket tid på dessa 4 områden. Vilket innebär:

Vecka 49:

Klasser

Vecka 50 - 52:

Encounter klassen

Vecka 52 - 2:

Meny

Vecka 2 - 4:

Gui

Vecka 4 - inlämning:

Gå igenom för att dubbelkolla allt fungerar

Jag kommer nog bli klar innan denna schemat, jag vill sikta på att bli klart innan jullovet är slut. Jag jobbar i gitbhub som jag dokumenterar min process. Där kommer jag även spara prototyper som jag kommer lämna in.